Available online at www.sciencedirect.com
SCIENCE DIRECT®
Ad Hoc Networks 2 (2004) 369–386
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Probabilistic reliable multicast in ad hoc networks ☆

Jun Luo *, Patrick Th. Eugster, Jean-Pierre Hubaux

School of Computer and Communication Sciences, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland

Received 7 May 2003; accepted 15 July 2003
Available online 4 September 2003

Abstract

When striving for reliability, multicast protocols are most commonly designed as deterministic solutions. Such an approach seems to make the reasoning about reliability guarantees (traditionally, binary, "all-or-nothing"-like) in the face of packet losses and/or node crashes. It is however precisely this determinism that tends to become a limiting factor when aiming at both reliability and scalability, particularly in highly dynamic networks, e.g., ad hoc networks. Gossip-based multicast protocols appear to be a viable path towards providing multicast reliability guarantees. Such protocols embrace the non-deterministic nature of ad hoc networks, providing analytically predictable probabilistic reliability guarantees at a reasonable overhead.

This paper presents the Route Driven Gossip (RDG) protocol, a gossip-based multicast protocol designed precisely to meet a more practical specification of probabilistic reliability in ad hoc networks. Our RDG protocol can be deployed on any basic on-demand routing protocol, achieving a high level of reliability without relying on any inherent multicast primitive. We illustrate our RDG protocol by layering it on top of the "bare" Dynamic Source Routing protocol, and convey our claims of reliability and scalability through both analysis and simulation.

Keywords: Ad hoc networks; Reliable multicast; Gossiping; Stochastic modelling

1. Introduction

Reliable multicast protocols are a main building block for distributed application development. Such protocols can be roughly divided into three categories in *wired* networks, according to the provided guarantees: (i) strict semantics with "all-or-nothing" delivery guarantees (e.g., [1]), (ii) practical reliability without quantitative guarantee (e.g., [2]), and (iii) probabilitic reliability (e.g., [3]). While the first category usually incurs very large overhead in order to tolerate transmission and node failures, and the second category can do with smaller footprint by only considering transmission failures, most protocols of the third category are tunable with respect to the tradeoff between overhead and reliability in the face of both transmission and node failures.

☆ This work was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (http://www.terminodes.org).

* Corresponding author.
E-mail addresses: jun.luo@epfl.ch (J. Luo), patrick.eugster@epfl.ch (P.Th. Eugster), jean-pierre.hubaux@epfl.ch (J.-P. Hubaux).

1570-8705/$ - see front matter
doi:10.1016/S1570-8705(03)00055-6

For ad hoc networks, [1] probabilistic schemes seem to be intuitively appealing, precisely because the underlying network itself provides little determinism: nodes are not connected by any fixed infrastructure, and communication between two such nodes at a given moment might be possible directly, only indirectly, or not at all. This intuition is also supported by the observation that deterministic protocols to multicast in ad hoc networks suffer strongly from an amplification of the contradiction between reliability and overhead already encountered with such protocols in wired networks. Existing (unreliable) protocols (ad hoc-analogues to *IP multicast* [4]) provide no reliability guarantees at all (e.g., [5,6]), and proposals attempting to detect and repair failures (e.g., [7,8]) can hardly generate any throughput when the network topology undergoes frequent changes. Furthermore, no protocols providing strong reliability guarantees in the sense of "all-or-nothing" semantics (the first aforementioned category of protocols) have yet been proposed due to the prohibitive complexity. In conclusion, a gossip-based probabilistic protocol can be a reasonable way to provide a form of multicast reliability in ad hoc networks. However, devising a gossip-based multicast protocol for ad hoc networks is not trivial and, in particular, cannot be straightforwardly achieved by adapting a protocol conceived for wired networks.

As a cornerstone in the *Terminodes* [9] project, this paper presents a novel gossip-based multicast protocol for ad hoc networks, designed to meet a more practical specification of probabilistic reliability. Our *Route Driven Gossip* (RDG) protocol (i) uses a pure gossip scheme, i.e., gossiping uniformly about multicast packets, negative acknowledgements, and membership information, (ii) takes into consideration parameters of the network, e.g., the availability of routing information, and (iii) does not require a multicast primitive at the network layer and can be deployed on any basic, virtually unmodified, on-demand routing protocol. We illustrate our RDG protocol using the "bare" Dynamic Source Routing (DSR) [10] protocol, i.e., without any multicast extension. We defend our claims of predictable reliability of the protocol by comparing results obtained through a formal analysis based on a stochastic model and results collected from an exhaustive set of simulation experiments performed with the *ns-2* network simulator [11]. The simulation results also confirm the scalability and adaptability of our protocol. The main idea of RDG is to explore the feasibility of such a probabilistic approach along with a prediction of its performance in a highly dynamic setting, useful for many critical applications such as security services (e.g., distributed key management services [12], or certificate distribution and revocation for self-organized public-key infrastructures [13,14]).

The rest of this paper is organized as follows: Section 2 surveys related work. Section 3 describes the network model and specifies more precisely the problem solved. Section 4 presents our RDG protocol. A formal analysis and simulation results of our RDG protocol are given in Sections 5 and 6, respectively. Section 7 discusses various issues, such as optimizations and reliability metrics. Finally, Section 8 concludes the paper.

2. Related work

This section summarizes previous work that are closely related to our proposal.

2.1. Deterministic reliable multicast in wired networks

In *wired* networks, reliable multicast protocols strive for strong, "all-or-nothing"-like, reliability semantics for the successful delivery of a message to a group of nodes despite the failure of a certain number of these nodes (cf. *Reliable Broadcast* [1]). These protocols scale poorly with an increasing group size even in a very stable network.

Protocols that indeed offer some practical reliability, but are not reliable in the metric of the above-defined category and lack an alternative measure of their reliability, include typically pro-

[1] Both mobile ad hoc networks and wireless sensor networks are considered here.

tocols building on top of IP multicast, such as [2,15]. The ack/nack mechanisms employed by such protocols to improve reliability, unfortunately, also tend to compromise their scalability by heavily loading the network (e.g., leading to *ack implosion*).

2.2. Gossiping in wired networks

Probabilistic multicast protocols are a family of protocols that has been rediscovered rather recently. Roughly, the basic idea is to have each node in a multicast group periodically "talk" to a random set of other nodes in the group about its knowledge of the "state" of the group, e.g., the multicast packets that it has received. Missing packets can then be recovered by nodes in a peer-based style (e.g., [3,16,17]). These protocols equally distribute the load over the nodes in a group and thus also make themselves very resilient to arbitrary node failures. Stochastic models derived from epidemiology enable the protocols to obtain (i) a performance prediction and (ii) the desired tradeoff between reliability and overhead by adjusting protocol parameters.

The *Probabilistic Broadcast* (*pbcast*) [3] protocol has in much rejuvenated the interest in gossip-based protocols that find their origins at Xerox where they were initially used for replicated database maintenance [18]. The *pbcast* protocol consists of two phases: a first phase based on an unreliable multicast primitive and a second one making use of gossips for repairing packet losses. These phases are merged into one phase by the *Lightweight Probabilistic Broadcast* (*lpbcast*) [17] protocol. By gossiping uniformly about data packets, digests, as well as membership information, *lpbcast* provides reliability similar to *pbcast* without imposing a complete membership view on the members.

Taking the network topology into account when gossiping, *Directional Gossip* (DG) [16] gains in efficiency. In short, a *weight* is computed for each neighbor node, representing the connectivity of that given node. The larger the weight of a node, the higher the possibility for it to receive a given packet from other nodes. When gossiping, nodes with higher weights are hence chosen with a smaller probability, reducing redundant sends. In particular, LANs are represented by single nodes to distant LANs, and "long" routes between two such representatives are seldom chosen.

While the DG protocol does not provide any analytical evaluation, protocols such as *pbcast* and *lpbcast* are analyzed in much detail based on a recurrence relation establishing the probabilities for the possible numbers of infected nodes at all gossip rounds. Alternatively, protocols are modelled by differential equations (e.g., [18]), or random graph theory (e.g., [19]). The latter protocol is tightly coupled to its analysis, in the sense that a particular packet is gossiped only once by a given node. Roughly, in such a model, there is a sharp threshold for the required fanout around $\log n$ (n being the number of members in a multicast group) to ensure that, with very high probability, all nodes will receive a given multicast packet despite node and transmission failures.

2.3. Gossiping in ad hoc networks

The benefits of gossiping techniques have, rather recently, also been exploited in ad hoc networks. In this context, gossip-based protocols are not favored for obtaining an analytical prediction of their performance in terms of reliability, but more for the practical observation that they (i) perform in a more reliable way than unreliable protocols such as MAODV [5] and (ii) generate less traffic than, for instance, flooding approaches.

Anonymous Gossip (AG) protocol [20], a descendant of the *pbcast* [3] protocol, pioneered the recent research efforts on gossip-based multicast protocols for ad hoc networks. Through the concept of anonymous gossip, any agreement on membership is avoided during the gossip-based repair phase. This however shifts the responsibility for the membership management to the MAODV layer, which the AG protocol also relies upon for a preliminary, rough packet dissemination. These prerequisites make the AG protocol more difficult to apply in a broader context than the one offered by MAODV. Furthermore, the property of predictable behavior, an important merit of

gossip-based protocols, is lost due to the dependence on MAODV to guide the gossips.

The exploitation of the observation (ii) is briefly mentioned in [21,22], and then more closely investigated by Haas et al. in [23] for the dissemination of routing messages. Since deterministic flooding techniques do not necessarily ensure that, in practice, every node sees a given information either, gossiping techniques yield results close to those of flooding protocols, yet imposing far less load on the network.

Prior to that, Vahdat and Becker [24] have also employed gossiping techniques for unicast routing. Their idea is to ensure that packets are eventually delivered even if there is no path between the source and the destination for some time. Such an approach is very interesting, but tends to require relatively high buffering capacities at individual nodes if all unicast traffic is handled that way. Just like all other gossip-based protocols for ad hoc networks we know of, this effort does not include any analytical performance estimation.

2.4. Stateless multicast

Last but not least, another recent paradigm shift is given by *stateless multicast* protocols [25,26]. While the *Differential Destination Multicast* (DDM) [25] protocol explicitly calls the unicasting function to disseminate multicast packets, the protocol presented in [26] builds an overlay multicast packet distribution tree on top of the underlying unicast routing protocol, and multicast packets are encapsulated in a unicast envelop and transmitted between the nodes in the group. While reducing the control overhead of the multicast session, the protocol leads to overweighted packet headers. This problem, known from unicast source routing but amplified in the case of multicasting, limits the protocol's scalability in terms of the group size.

3. Assumptions and problem

Before presenting our RDG protocol, we define more accurately our network model and specify the problem solved in that model.

3.1. Network model

The network consists of a set $\mathbb{N}$ of nodes with the same computation and transmission capabilities, communicating through bidirectional wireless links between each other. A unicast routing protocol is available to support packet transmissions between the network nodes (we assume DSR in this paper). $G \subset \mathbb{N}$ is a multicast group with size $|G| = n_G$. Nodes join and leave different groups following the requirements of upper layer applications.

The following assumptions are made on the nodes:

- Every node has a unique physical address or ID.
- The transmission radius for each node is fixed.
- Nodes fail only by crashing, i.e., stopping to function. Crashes are not permanent.

In addition, we assume a CSMA/CA-like MAC layer protocol (e.g., IEEE 802.11) that provides a RTS/CTS-Data/Ack handshake sequence for each transmission.

The information unit for the protocol is the *message*. It can include data packets, as well as membership information. However, the *packet*, the unit for the network layer, is used when data logging and loss detection are carried out. Each packet multicast is uniquely identified by its identifier *pid*, a tuple (group ID, source ID, packet sequence No.), such that a member can detect missing packets by observing gaps in the packet ID sequence.

3.2. Problem definition

Our goal is to design a multicast protocol for ad hoc networks, which achieves probabilistic reliability. Instead of providing perfect guarantees like "all packets sent by a source will eventually be received by all correct group members", we provide one that roughly states "if some group member sends out a flow of M packets, a certain group member receives a fraction $\zeta \leqslant x$ of all M packets with probability $\pi_M(x)$ $(\zeta, \pi_M \in [0,1])$". Here ζ and π_M are termed *reliability degree* and

reliability probability distribution, [2] respectively. The reliability of the protocol defined by $\pi_M(x)$ is expected to be predictable given simple information like packet loss ratio, whereas the scalability requirements are such that increasing network size and mobility should only result in a modest degradation of reliability.

4. Route Driven Gossip protocol

This section presents in detail our RDG protocol after providing related background.

4.1. Overview of DSR protocol

Dynamic Source Routing is an on-demand routing protocol making use of source routing and an aggressive caching policy. The protocol is on-demand since it floods route requests in the network upon routing packets to a destination without an available corresponding routing path. The source routing mechanism makes the routing paths loop-free, while providing certain topological information. With the aggressive caching policy, DSR tries to cache all routing paths that it learns (it even taps such information from the MAC layer if the "promiscuous" receive mode is enabled.).

4.2. Design rationale

Traditional gossip protocols are characterized as *view driven gossip* because the destinations of each gossip are determined by the view [3] of the membership at the source. According to our observations, a view driven protocol is unsuitable for ad hoc networks with on-demand routing (e.g., DSR and AODV), since a node cannot always have routing paths to all the nodes in its view. If each node would request the paths to its gossip destinations for each gossip task, heavy network traffic would be generated, reducing the efficiency of the protocol. In addition, our problem definition deviates from those of traditional gossip protocols by considering the dissemination of a continuous flow of packets.

The design of our gossip-based protocol has been influenced by the following observations on ad hoc networks working with an on-demand routing protocol:

- *Routing information is precious*, because the costs to obtain such information are considerably high. In our case, the routing information for group members covers not only the routing paths but also the links between a certain member and its routing paths. It is possible that either there is no routing path to a known member or an existing routing path leads to a member that is unknown to the source. In order to make the best of these resources, the protocol should maintain as many links as possible and try to use them while they are fresh.
- *Route requests are costly*, due to the propagation of route requests with flooding. We can, however, benefit from this feature by requesting the routing paths to several group members with only one request message. Although the network traffic is greatly reduced in the request phase, the massive reply messages in the reply phase afterward may congest the network. Therefore, one still needs to be careful in dealing with the route reply.
- *Each group member is aware of the packet losses*, given the *pid* sequence of received packets. A protocol can exploit this feature to enhance its reliability without incurring too much overhead.

4.3. Protocol presentation

In order to overcome the problems with *view driven* protocols in ad hoc networks and to integrate the observations stated above, we propose a *route driven* protocol. Our RDG protocol relies only on partial views for each member; these random subviews result from the randomness of routing information that nodes can have. RDG uses a *pure* gossip scheme. The spread of the

[2] π_M is actually the *cumulative distribution function* (cdf) of ζ.

[3] *View* is a data structure to store the membership information.

information is propelled mainly by a *gossiper-push* (each group member forwards multicast packets to a random subset of the group) but complemented by a *gossiper-pull* (multicast packets piggyback negative acknowledgements of respective forwarding group members).

4.3.1. Basic data structures

There is one protocol instance for each group G. Besides the identifier of a group (*Gid*), the following four data structures are used for the protocol:

- **Data buffer** (*Buffer*): This buffer stores data packets received. It is divided into two parts: *Buffer.new* stores the packets to be gossiped in the future; the other packets are stored in *Buffer.old* in preparation for responding to gossiper-pulls. If the size limit of the buffer is reached, the oldest packets are removed.
- **Active view** (*AView*): This view contains the IDs of known members to which at least one routing path is known.
- **Passive view** (*PView*): Contains the IDs of known members to which no routing path is currently available.
- **Remove view** (*RView*): Contains the IDs of members that have indicated their desire to leave.

Therefore, each $node_i \in G$ has five data structures: Gid_i^G, $Buffer_i^G$, $AView_i^G$, $PView_i^G$, and $RView_i^G$.

4.3.2. Operations

Our RDG protocol offers seven operations, which are grouped into three sessions corresponding to their functionality. The *join* session defines the behavior of the node interested in joining a group and the reactions of other group members. The *leave* session defines the behavior of the node intending to leave the group and the reactions. The GOSSIP task is periodically executed by a node (if there are messages to disseminate). Furthermore, nodes react to the gossip messages received. In relation to the GOSSIP task, two protocol parameters are defined here: the *fanout* (F) is the number of gossip destinations randomly selected from the *AView* for each gossip emission;

procedure JOIN(gid)
  GROUPREQUEST(id_i, gid)

upon RECEIVEGROUPREPLY(id, gid) **do**
  $AView_i^{gid} \leftarrow AView_i^{gid} \cup \{id\}$

(a) *Join* indication emission and reply reception

upon RECEIVEGROUPREQUEST(id, gid) **do**
  for all group G that i belongs to **do**
    if $Gid_i^G = gid$ **then**
      $AView_i^{gid} \leftarrow AView_i^{gid} \cup \{id\}$
      GROUPREPLY(id_i, gid) with probability P_{reply}

(b) *Join* indication reception

Fig. 1. *Join* session at node i.

the *quiescence threshold* (τ_q) is related to each data packet: a packet will be removed from *Buffer.new* after having been gossiped for τ_q times. Section 6.2 discusses how to set these parameters.

We extend the ROUTEREQUEST and ROUTEREPLY primitives provided by DSR for our purposes:

- GROUPREQUEST (Fig. 1(a)): This primitive extends the ROUTEREQUEST of DSR by requesting routing paths to multiple nodes at the same time. The GROUPREQUEST puts the group ID (*gid*) in the target address field of the DSR header. Only group members can respond to the message.
- GROUPREPLY (Fig. 1(b)): This primitive is equivalent to a ROUTEREPLY but with the *gid* attached to it, such that a node receiving such a message can distinguish it from a usual ROUTEREPLY.

The RDG protocol can be rather easily adapted to other on-demand routing protocols by accordingly implementing these primitives.

4.3.3. Protocol behavior

The pseudo-codes for all the above operations are provided here, followed by detailed descriptions. The *gossip* and *leave* sessions are reported together, since the dissemination of a leave indi-

cation relies on the *gossip* session. Note that data structures like *Buffer* have a maximum size, noted $|L|_m$ for a given list L.

Join session (Fig. 1):

- A node intending to join a group floods the network with a GROUPREQUEST message to search for other group members whilst announcing its existence.
- Upon receiving a GROUPREQUEST from a certain member, all members update their *AView* with the new ID. They also return a GROUPREPLY to the request initiator with probability P_{reply}.
- The initiator of the GROUPREQUEST also updates its *AView* after receiving the GROUPREPLY.

By recording the route of each incoming packet, DSR ensures that a new element in *AView* has a corresponding route entry in the DSR routing table. The validity of this link is periodically checked and the *AView* and *PView* are updated accordingly. When the size of *AView* drops below some threshold τ_v, the node has to reinitiate a *join* session.

Gossip/leave session (Figs. 2 and 3)

- Each member of the group periodically (every T ms) [4] generates a gossip message and gossips it to F other nodes randomly chosen from *AView*. The message includes packets stored in *Buffer.new*, and the *id* of the most recent missing packet. It also piggybacks its view on the membership (if the node intends to leave, only the field of *RView* is valid). A data packet is removed from *Buffer.new* after having been gossiped for τ_q times.
- A group member receiving a gossip message will (i) remove the obsolete member from its view, (ii) add the new member to the view, (iii) update the data buffer with new packets, and (iv) respond to the gossiper-pull. The gossiper-pull is responded to only if the data packet requested will not be gossiped again (the request might be satisfied by the upcoming gossip).
- A packet received upon gossiper-pull is delivered if it is still missing. The data buffer is updated accordingly.

```
procedure LEAVE(gid)
  leaveFlag_i^gid ← true

task GOSSIP(gid)                    {/* Executed every T ms */}
  /* Step 1: Generate message and disseminate it */
  if leaveFlag_i^gid = true then
    m.rview ← id_i
  else
    m.data ← Buffer_i^gid.new
    m.gpull ← pid of the most recent missing packet
    /* gossiper-pull */
    m.rview ← a random entry in RView_i^gid
    m.view ← a random entry in AView_i^gid ∪ PView_i^gid
  DS ← random set of F members in AView
  for all id ∈ DS do
    SENDGOSSIP(gid, id_i, m, id)

  /* Step 2: Update the data buffer */
  if leaveFlag_i^gid = false then
    if ∃ pkt ∈ Buffer_i^gid.new that has been gossiped
    more than τ_q times then
      Buffer_i^gid.old ← Buffer_i^gid.old ∪ {pkt}
      Buffer_i^gid.new ← Buffer_i^gid.new\{pkt}
```

Fig. 2. *Gossip/leave* session at node i—message emission.

Nodes along routing paths to gossip destinations belonging to the same group as those destinations, when forwarding a packet they have not received yet, also deliver the packet and update their data buffers (not shown in the code). Due to its unpredictability, this operation will not be taken into account in the analysis in the next section, making the protocol perform better than expected.

Note that the packet salvaging function of DSR is disabled while a gossip message is on its way, i.e., packets are dropped immediately whenever the routing path becomes obsolete or the sending buffer overflows. In fact, the redundancy provided by our RDG protocol automatically offsets the packet loss.

[4] In order to save bandwidth, we apply the *binary exponential backoff* algorithm to adjust the period when there is no new packet to be sent or no lost packet to be requested.

```
upon RECEIVEGOSSIP(gid, id_s, m) do
  /* Step 1: Remove obsolete member from the view */
  AView_i^gid ← AView_i^gid \ {m.rview}
  PView_i^gid ← PView_i^gid \ {m.rview}
  RView ← RView ∪ {m.rview}
  while |RView| > |RView|_m do
    remove a random element from RView

  /* Step 2: Add new member to the view */
  if m.view ∉ (AView_i^gid ∪ PView_i^gid) then
    if there exists a route to that node then
      AView_i^gid ← AView_i^gid ∪ {m.view}
    else
      PView_i^gid ← PView_i^gid ∪ {m.view}

  /* Step 3: Update Buffer with new packets */
  for all pkt ∈ m.data do
    if pkt ∉ Buffer_i^gid then
      Buffer_i^gid.new ← pkt
      DELIVER(pkt)        /* to the upper layer */
  while |Buffer_i^gid| > |Buffer_i^gid|_m do
    remove the oldest element from Buffer

  /* Step 4: Respond to the gossiper-pull */
  if m.gpull ∈ pid list of Buffer_i^gid.old then
    SENDGOSSIPRESPONSE(gid, id_i, pkt_{m.gpull}, id_s)

upon RECEIVEGOSSIPRESPONSE(gid, id, pkt) do
  if pkt ∉ Buffer_i^gid then
    Buffer_i^gid.old ← pkt
    DELIVER(pkt)        /* to the upper layer */
```

Fig. 3. *Gossip/leave* session at node i—message reception.

4.3.4. Topology-aware RDG

The basic RDG protocol presented above can be qualified as a *brute force* protocol. It can be made aware of the network topology for improved efficiency. We call the variant TA-RDG, i.e., topology-aware RDG. The design of this variant is based on the assumption that the underlying routing protocol can provide some partial topological information, e.g., we can have the information about the lengths of paths from the routing table of DSR. The heuristics based on DSR work like this: different weights are assigned to the members in *AView* according to the length of the routing paths to them, i.e., the longer the path the lower the weight, such that a node directs a gossip message towards a "near" member with higher probability. A simple way to implement this is to choose weights inversely proportional to the length of the corresponding routing paths. The locality of the traffic resulting from this optimization greatly reduces the network load and, as shown by simulations, improves the reliability in most cases.

4.4. Example of protocol operation

We assume a single group G of size $n_G = 10$ within a 20 nodes network. Fig. 4 gives a visual illustration of the behavior of the protocol with respect to the dissemination of one packet. Assuming $F = 2$ and $\tau_q = 2$, the packet initiated by member 15 infects the whole group in only three rounds in spite of the fact that no member has a full view of the membership while nodes move and even fail. By comparison, Fig. 5 illustrates the behavior of the protocol with respect to the dissemination of two consecutive packets, assuming $F = 2$ and $\tau_q = 1$. Note that the strength of gossiper-pull becomes evident. The figures intuitively show that using gossiper-pull is a cheaper way to improve the protocol reliability than having $\tau_q \geqslant 1$ in the case of continuous packet dissemination; this intuition is proven in Section 6.

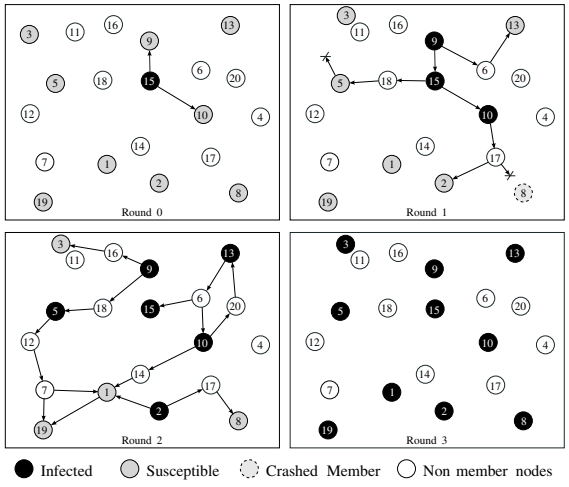

Fig. 4. An example of one "run" of the protocol with $F = 2$ and $\tau_q = 2$ within a group of size 10. A member may receive duplicates of the same packet (e.g., member 1 at round 2). On the other hand, the packet can get lost at a certain round due to nodes crashing or moving (e.g., members 8 and 3 in round 1), but these losses will be compensated with high probability at a later round.

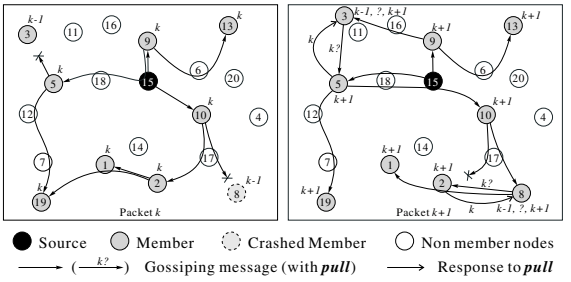

Fig. 5. An example of two "run"s of the protocol with $F = 2$ and $\tau_q = 1$ within a group of size 10. The likelihood of receiving duplicates of the same packet is reduced due to the smaller value of τ_q, which implies a lower overhead, but at the cost of a reliability degradation in a run; this cost is, however, compensated in the next run.

5. Analysis

This section provides an analytical evaluation of our RDG protocol (however, without considering the topology-awareness, in order to simplify the tractability). The goal is to show that the reliability of the protocol is predictable given certain protocol parameters and information about the network. This claim is confirmed by simulations in the next section.

5.1. Model

We consider a single group G composed of $|G| = n_G = n$ members and observe its behavior in terms of the dissemination of a *single* packet ("one run"), but also a *continuous* flow of packets (which is more practical than related efforts considering only the "one run" part). According to the terminology of epidemiology [27], a member that has received a certain packet is termed *infected*, otherwise *susceptible*. An infected member attempting to share the packet with others (i.e., a member who keeps gossiping the packet) is called *infectious*.

We analyze our protocol in a network composed of a static set of nodes running closely synchronized. More precisely, nodes gossip in synchronous rounds (T ms, identical for all nodes), and there is an upper bound on the network latency which is smaller than T.

The probability of packet loss is closely related to the movement and traffic pattern, as well as to the length of the considered routing path. By assuming an identical and independent probability of failure p_f for each hop along a routing path in a certain network environment, the probability of losing a certain gossip message can be expressed as a function of the number of hops, H, of that routing path. We further assume that the lengths H of all routing paths between two members follow the same distribution $f(h)$. On the other hand, p_f can be split into two parts: (i) p_{f_c} represents the probability of packet loss due to node crash; (ii) $p_{f_{mo}}$ accounts for the effects of node mobility and buffer overflow. While p_{f_c} can be set according to empirical results, $p_{f_{mo}}$ is determined by the movement and traffic pattern. Furthermore, we assign a probability p_{nc} to each member, in order to characterize its possible non-cooperative behavior (i.e., a member declines to forward a packet with probability p_{nc}).

In reality, the size of the *AView* for a given member may vary between τ_v and $n - 1$. However, the value could be maintained very close to $n - 1$ by assuming a low mobility network. Furthermore, we expect that the protocol can keep approximately the same view size in a high mobility network, assuming that other protocols running in parallel infuse routing information to the nodes.

Due to its irregularity, the effects of the gossiper-pull procedure can hardly be considered in the analysis, making the present analysis a lower bound.

5.2. Stochastic behavior of packet dissemination

The predictable reliability of our RDG protocol is conveyed in two steps. We first show that the *single packet dissemination reliability* is predictable given certain network information, and based on the results, we discuss the *reliability probability distribution* π_M.

5.2.1. Single packet dissemination reliability

Let m be a message generated by a certain member. We use $S_r \in \{1, \ldots, n\}$ and $\Delta S_r = \mathbf{E}[S_r - S_{r-1}]$ to denote the number of members

infected with m *after* round r and the average [5] number of members infected *within* round r, respectively. If we define the *state space* $\mathscr{E} = \{1, \ldots, n\}$, the sequence of random variables $\{S_r\}_{r \geqslant 0}$ forms a stochastic process with values taken from $\mathscr{E}$.

(a) *Recurrence relation*: Given the probability p that a certain member is infected by a gossip message, $q = 1 - p$ represents the probability of non-infection. With $S_r = i$ (the number of infected members) and $\sum_{t=1}^{\tau_q} \Delta S_{r+1-t} = \delta$ (the number of infectious members) in the current round, we introduce a binary random variable, X_k, for each of the remaining $n - i$ susceptible members, where $\Pr\{X_k = 0\} = q^\delta$, i.e., the probability that a certain susceptible member is not infected in the next round is the probability that it is not infected by any of the δ infectious members. It is clear that $S_{r+1} - S_r = \sum X_k$ follows a binomial distribution. For a given number of j infected members in the next round, the transition probability $p_{(i,j)\delta}$ is expressed as

$$\begin{aligned} p_{(i,j)\delta} &= \Pr\left\{ S_{r+1} = j | S_r = i, \sum_{t=1}^{\tau_q} \Delta S_{r+1-t} = \delta \right\} \\ &= \Pr\left\{ \sum X_k = j - i | \mathbf{E}[X_k] = 1 - q^\delta \right\} \\ &= \begin{cases} \binom{n-i}{j-i} (1 - q^\delta)^{j-i} q^{\delta(n-j)}, & j \geqslant i, \\ 0, & j < i. \end{cases} \end{aligned} \tag{1}$$

Then, with the convention that message m is injected into the system at round $r = 0$ by the originating member, the initial distribution of S_r is given by

$$\Pr\{S_0 = j\} = \begin{cases} 1, & j = 1, \\ 0, & j > 1. \end{cases} \tag{2}$$

Having the initial distribution and transition matrix $\mathscr{P}_\delta = \{p_{(i,j)\delta}\}_{i,j,\delta \in \mathscr{E}}$, v_r, the distribution of S_r, is then computed as

$$v_{r+1}^{\mathrm{T}} = v_r^{\mathrm{T}} \mathscr{P}_\delta, \tag{3}$$

where $v_r(i) = \Pr\{S_r = i\}$ is the ith element of the column vector v_r.

(b) *Determining parameters*: According to our assumptions, the probability p can be estimated by taking three conditions into account: (i) the gossip source is cooperative, (ii) the considered node is chosen as the gossip destination and (iii) the gossip message is successfully received. This results in the following expression:

$$\begin{aligned} p &= \overbrace{(1 - p_{nc})}^{\text{(i)}} \overbrace{P_{gossip}}^{\text{(ii)}} \overbrace{P_{succ}}^{\text{(iii)}} \\ &= (1 - p_{nc}) \left(\frac{F}{n-1} \right) P_{succ}. \end{aligned} \tag{4}$$

Given a certain length (in hops) h of a routing path, the probability of successful delivery is expressed as $P_{succ} = (1 - p_f)^h$. According to Bayes's rule of exclusive and exhaustive causes [28]:

$$\begin{aligned} P_{succ} &= \sum_h (1 - p_f)^h \Pr\{H = h\} \\ &= \mathbf{E}_H[(1 - p_f)^H]. \end{aligned} \tag{5}$$

Therefore, p is expressed as

$$p = (1 - p_{nc}) \left(\frac{F}{n-1} \right) E_H[(1 - p_f)^H]. \tag{6}$$

The distribution of H and the value of p_f are the network information we need. We discuss their estimations in the Appendix.

5.2.2. Reliability probability distribution

Having the single packet dissemination reliability measure $v(i)$ [6], the reliability of disseminating a flow of M packets, i.e., $\pi_M(x)$, can be expressed as

$$\pi_M(x) = \sum_{i=0}^{\lfloor Mx \rfloor} \binom{M}{i} p_1^i (1 - p_1)^{M-i}, \tag{7}$$

where $p_1 = \sum i \cdot v(i)/n$ is the probability that a certain group member receives a single packet. Here we assume that the receptions of two distinct packets are independent events.

[5] Setting $\Delta S_r = S_r - S_{r-1}$ would make ΔS_r a random variable, leading to a state space unfeasible for analysis. Our approximation results in simplified calculations without sacrificing too much of the mathematical rigor.

[6] The subscript r is dropped hereafter, because we always consider the final distribution after the last round.

6. Simulations

This section presents the practical evaluation of our RDG protocol. We first compare our simulation results with the corresponding analytical ones in order to confirm the predictability of our RDG protocol. We then evaluate the advantage of TA-RDG against RDG by showing the improved protocol efficiency with the metric defined in [29]. Moreover, we show the sensitivity and adaptability of TA-RDG to the increasing fraction of non-cooperative members. Finally, we compare the reliability of TA-RDG with the Anonymous Gossip [20] protocol. [7]

6.1. Model

The version of *ns-2* we have made use of includes the Monarch Project wireless and mobile extensions. Besides various implementations of ad hoc routing protocols, e.g., DSR, the Monarch extensions incorporate a radio model based on the Lucent WaveLAN IEEE 802.11 product, which provides a 2Mbps transmission rate and a nominal range of 250 m. We adopt the two-ray ground reflection model as the radio propagation model.

We simulated a mobile ad hoc network with 100–200 nodes in a 1000 m × 1000 m area, operating over 360 s of simulated time. The movement pattern was defined by the random waypoint model. Each node had a maximum speed between 2–20 m/s and an average pause time of 40 s.

The network contained a single multicast group. Beginning at 10 s, the members consecutively joined the group until around 60 s. Then one of the members started to generate constant bit rate (CBR) traffic at regular intervals of 200 ms with each packet having a length of 64 bytes until 340 s. All nodes left the group at 350 s. The gossip period was also set to 200 ms. Each simulation was carried out 10 times with different scenario files created by *ns-2*.

[7] Comparisons with AG on efficiency are desirable but infeasible due to the big differences between the assumptions about the underlying mechanisms.

6.2. Comparing analytical and simulation results

Fig. 6(a) and (b) shows comparison between the analytical and simulation results of the basic RDG protocol, which are carried out by contrasting the evolution of the infection processes. These comparisons basically prove that the theoretical prediction of the relationship between the reliability and the latency is valid.

It is easy to observe that the reliability of the protocol with $F = 3$ is higher than the one with $F = 2$, because the fanout has a significant effect on the reliability. However, when we further increase the fanout, the reliability decreases rather than increases (analysis) or only marginally increases (simulation). The reason is that increasing the fanout has the same effect as increasing the number of connections, and p_f increases dramatically due to the network congestion. A similar reason accounts for what happens when τ_q changes from 1 to 2.

In fact, there is always a tradeoff between certain requirements on reliability and the introduced overhead, characterized by the values of F and τ_q. Considering the network capacity imposes a further limitation not considered in other efforts (considerably large F [19] or unbounded τ_q [17]).

Fig. 6(c) and (d) shows the reliability of both RDG with and without gossiper-pull for different mobility patterns and group sizes. We provide here the mean value of ζ and its standard deviation, which characterize the distribution function π_M. The figures again exhibit the similarity between the simulation and analytical results with respect to RDG without gossiper-pull. As expected, RDG with gossiper-pull always performs better than RDG without gossiper-pull, while the improvement is significant in high mobility and large group scenarios. We also note that only a slight reliability degradation is observed when the mobility or group size is increased, illustrating the scalability of RDG.

6.3. RDG versus TA-RDG

Fig. 7 shows the reliability and overhead of both our basic protocol (RDG) and its variant

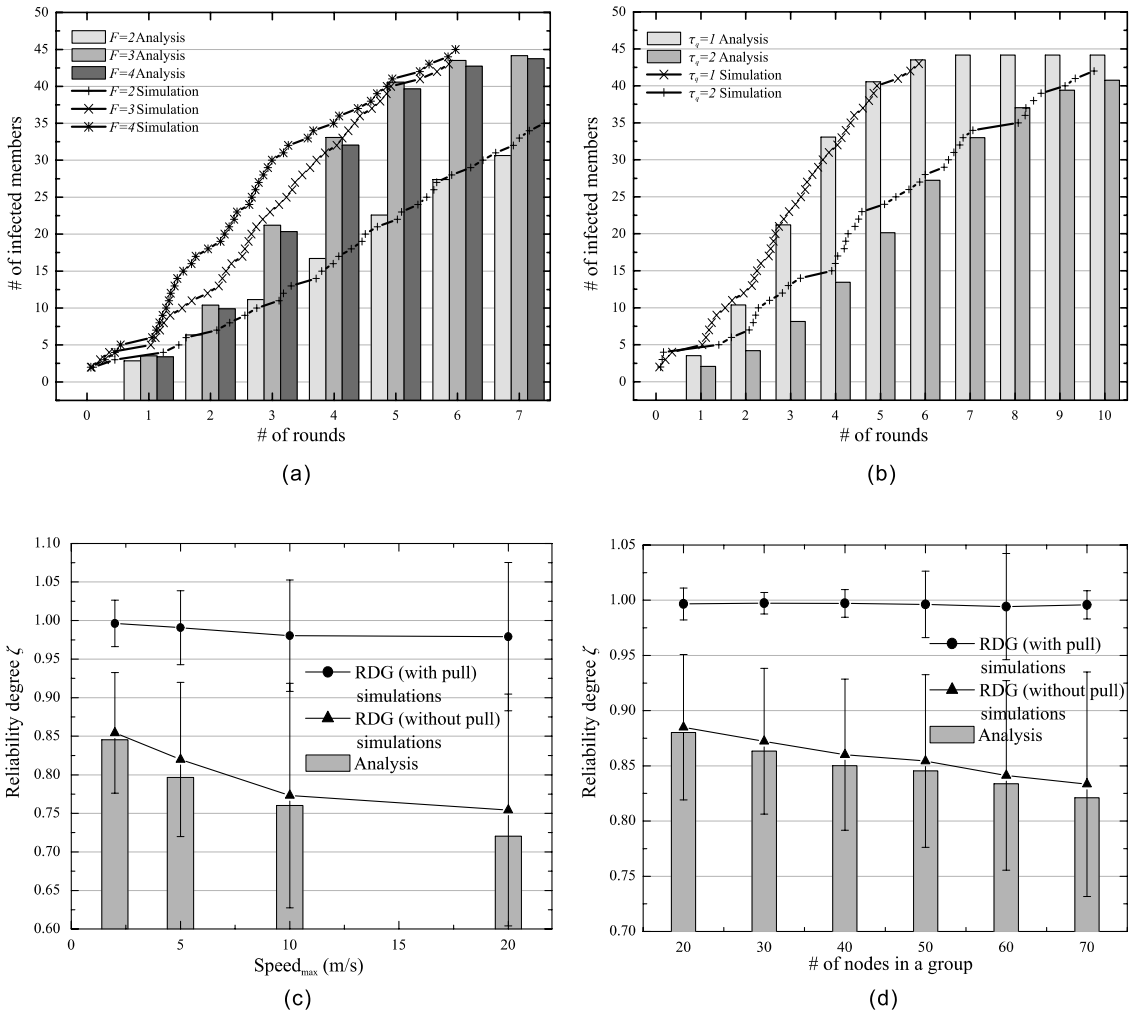

Fig. 6. Comparison between analytical and simulation results within networks of 100 nodes. Single packet dissemination: (a) $\tau_q = 1$ and $F = 2, 3, 4$, groups of 50 nodes, $speed_{max} = 2$ m/s; (b) $\tau_q = 1, 2$ and $F = 3$, groups of 50 nodes, $speed_{max} = 2$ m/s. Continuous packet dissemination: (c) groups of 50 nodes, $\tau_q = 1$ and $F = 3$; (d) $speed_{max} = 2$ m/s, $\tau_q = 1$ and $F = 3$.

(TA-RDG) with different mobility patterns, group sizes, and network densities. For the simulations in this subsection, we always assume a group containing half of the network nodes. The overhead is measured by the *network load* (defined in [29]) that takes into account, for each multicast, the number of unicast packets sent and the number of hops traveled by each packet. The results show that TA-RDG performs better than RDG in all cases, with respect to both reliability and overhead. The improvement of reliability is significant for large groups in high density networks, while the reduction of overhead is evident for low mobility scenarios in low density networks.

6.4. Sensitivity and adaptability to the increasing p_{nc}

In Fig. 8, we first show the sensitivity of TA-RDG to the increasing p_{nc} by fixing $F = F_0 = 3$ (see the bottom curves). The reliability degree ζ degrades modestly for small value of p_{nc} (i.e., $p_{nc} = 0.1$ and 0.2). At the same time, it is observed that the larger the fraction of non-cooperative members is, the lower the network load becomes

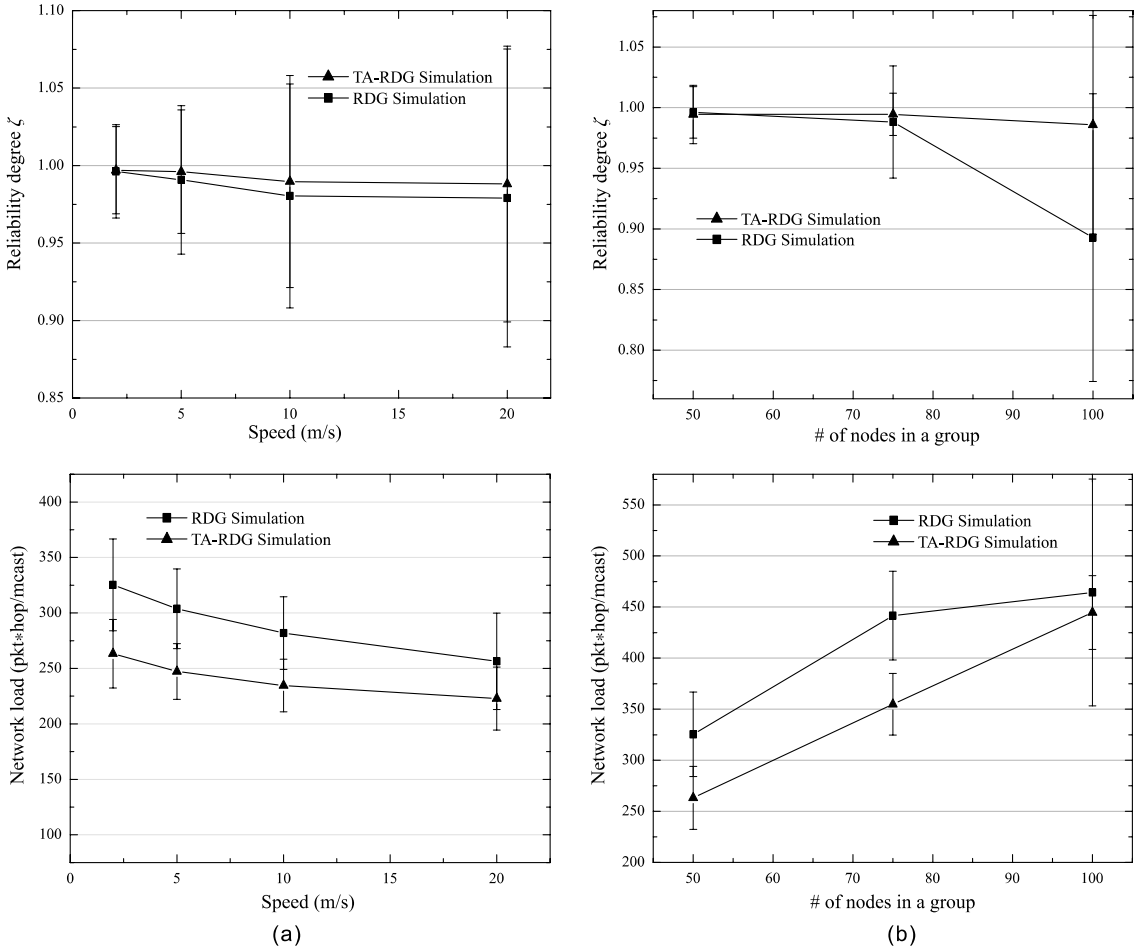

Fig. 7. Comparison between RDG and TA-RDG in terms of reliability and overhead, for different mobility patterns and network densities. (a) Groups of 50 nodes, $\tau_q = 1$ and $F = 3$; (b) $speed_{max} = 2$ m/s, $\tau_q = 1$ and $F = 3$.

(because a non-cooperative member does not forward a packet). Therefore, non-cooperative members become "beneficial" to the protocol if the reliability degradation is tolerable. However, the reliability degree decreases dramatically when we further increase p_{nc} (i.e., $p_{nc} = 0.3$ and 0.4). In these cases, the protocol has to adjust itself to cope with the situations. The principle we applied for the adjustment is based on (6): increase the fanout such that $F_{p_{nc}} = F_0/(1 - p_{nc})$, in order to keep p invariant. For example, $F_{0.3} \approx 4.3$. [8] The simulation results (see the upper curves) show that the adjustment leads to a tolerable degradation of reliability even when $p_{nc} = 0.4$, while incurring a small increment of overhead.

6.5. Comparing AG and TA-RDG

A systematic comparison between TA-RDG and AG [20] (discussed in Section 2.3) is hard, due to their different design goals. We compare them in the context of small groups, which should actually favor AG since RDG is designed for larger groups. The comparison is done by superimposing a figure from [20] with corresponding simulation results for RDG (for the same

[8] A real number $x.y$ for F means that each member, when forwarding a packet, takes $F = x$ with probability $1 - y/10$ and $F = x + 1$ with probability $y/10$.

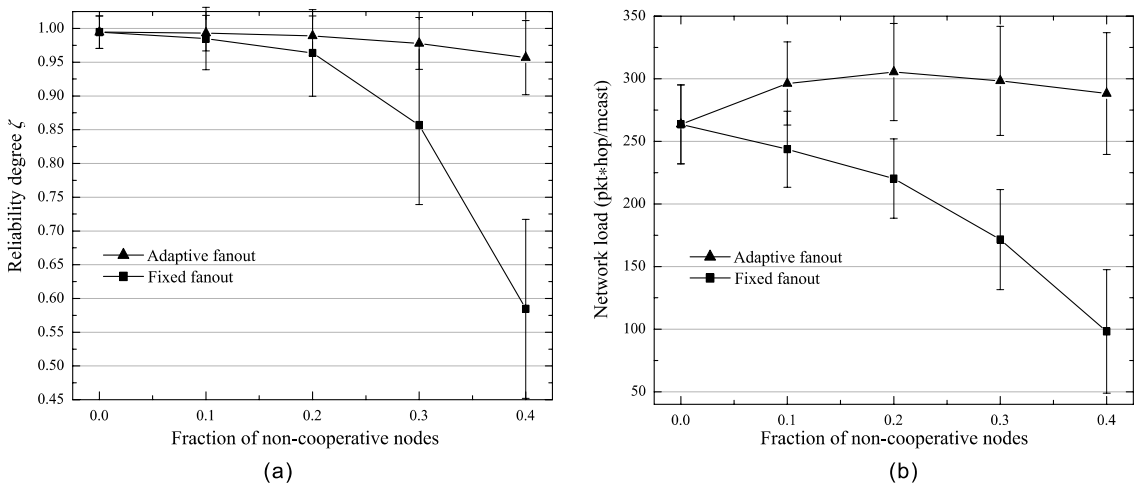

Fig. 8. The performance of TA-RDG under different fractions of non-cooperative members, with $n = 50$ and $speed_{max} = 2$ m/s in 100 node networks. (a) Reliability degree and (b) network load.

scenario). The figure (Fig. 9) shows that RDG is more reliable than AG in most cases. Furthermore, AG cannot compete with RDG in terms of scalability because it is based on the underlying multicast protocol whose overhead is much larger than the one of the unicast protocol that RDG is based on. Finally, the reliability of the AG protocol is not as predictable as RDG's is, since it relies on the existence of an unpredictable multicast tree.

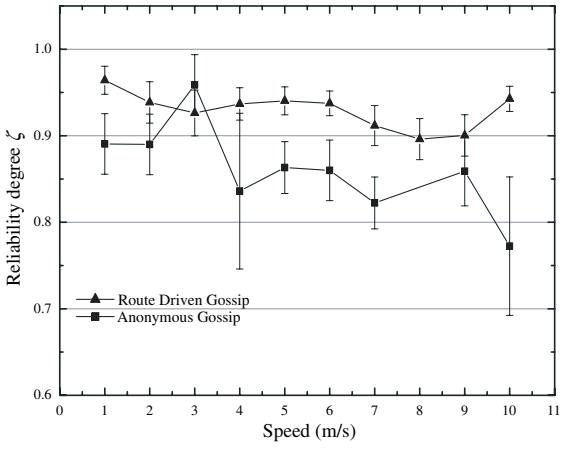

Fig. 9. Reliability of the AG and RDG protocols in a network of 40 nodes with approximately one-third of them in a group, located within a square of 200 m × 200 m. The maximum node speed varies between 1–10 m/s and the average pause time is 40 ms. The transmission range is 75 m.

7. Discussion

In this section, we discuss the possibility of evaluating our RDG protocol with an alternative specification to ζ and π_M as well as potential optimizations of RDG.

7.1. Protocol evaluation against Δ-reliability

Based on the previous analysis and the protocol description, we also evaluate here the reliability of our RDG protocol in the face of another specification defined in [30] consisting of the following three properties:

- *Validity*: correct process p multicasts $m \Rightarrow p$ delivers m. This can be trivially shown based on the protocol description.
- *Integrity*: m is delivered at most once for each correct process p, and only if $sender(m)$ multicasts m before. Since we do not consider Byzantine failures, no packet will be generated from the air. However, due to the limitations of buffers holding digests, the uniqueness of delivery

is, despite unique packet identifiers, hard to ensure infinitely. This problem is not unique to our approach, especially since packet identifiers are not unbounded, but are reused. In practice, buffers have however proven sufficient capacity to avoid the observation of any duplicate delivery.

- *Agreement*: correct process p delivers $m \Rightarrow a$ fraction ρ of correct processes deliver m with probability ψ. By taking $\rho = s/n$, our protocol satisfies this property with probability $\psi(\rho) = v(s)$.

7.2. Optimizations

The following are some optimization heuristics. The reason that we do not apply them to our protocol at this stage is that, although possibly improving performance, they somewhat put the randomness embedded into the protocol at stake, making performance prediction hard.

- Use multicast to disseminate gossip messages. By exploiting the potential multicast support provided by DSR, the gossiper node builds a source tree based on the available routing information. Only one message is transmitted through a certain tree edge. Different copies of the message are generated only at the bifurcation node.
- Assign P_{reply} adaptively at each member depending on the distance to the initiator of the GROUPREQUEST, i.e., the longer the path the bigger the value for P_{reply}. If a "near" member receives a GROUPREPLY from a "distant" one after it decides not to reply to the GROUPREQUEST, it would append its own reply to the packet before forwarding it. This optimization reduces the probability of different members along the same path separately generating a ROUTEREPLY, and hence reduces the bandwidth consumption.
- Add a directional flavor to the gossip scheme. A node would carefully select the directions of the gossip by directing the message to target peripheral members, i.e., the members that might not receive the gossip message in the current round, according to the knowledge of this node on the gossip messages it receives. The awareness of direction could be obtained by a GPS system, but also by a GPS-free mechanism (e.g., [31]).

8. Conclusions

In this paper we have presented a probabilistic approach to multicast, including a non-binary specification of multicast reliability and a gossip-based protocol, called Route Driven Gossip, conforming to this specification.

After comparing our approach with related work, we have described the operations of our RDG protocol, and developed an analysis of its performance, based on which the parameters (fanout and quiescence threshold, notably) can be fine tuned; we have shown the rapid propagation of data to all reachable members of the group; we have confirmed these results by simulations. Through this case of reliable multicast, we have illustrated that probabilistic approaches are indeed well suited for the challenging peculiarities of ad hoc networks.

In the near future, we intend to optimize our RDG protocol with respect to its overhead. We expect this to help us improve the practicality of RDG, in the sense of the modest cost incurred by the added reliability. This might give an indication on how our RDG protocol could be used by upper layer applications in an efficient way.

Acknowledgements

The authors would like to thank Milan Vojnovic, Rachid Guerraoui, Mario Cagalj, and Catherine Rosenberg for several instructive discussions.

Appendix

In order to obtain the distribution of H, we assume that the network nodes are uniformly distributed within a circle of diameter equal to

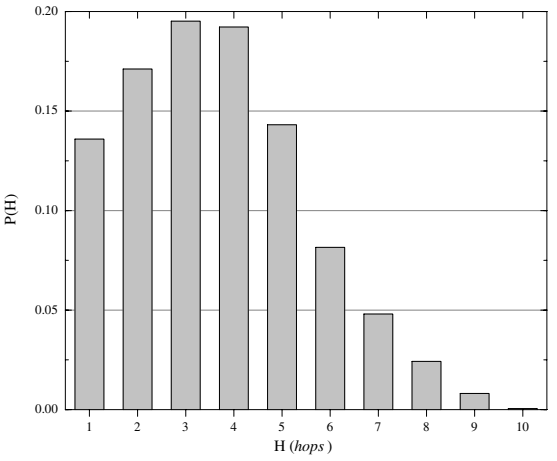

Fig. 10. Distribution of H. Here H is the random variable representing the distance between two randomly picked points within a circle. It can be considered as the length in hops of a routing path between two randomly picked network nodes, with the assumption that the nodes are uniformly distributed.

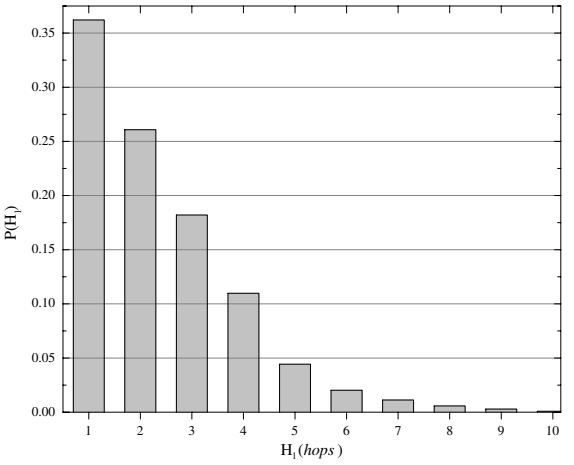

Fig. 11. Distribution of H_l when average packet loss ratio equals to 12.7%, assuming a group size of 50 and a network size of 100 with $F = 3$ and $\tau_q = 1$.

F \ τ_q	1	2
2	0.0200	—
3	0.0460	0.2749
4	0.1686	—

Fig. 12. The $p_{f_{mo}}$ with respect to different values of F and τ_q, assuming a group size of 50 and a network size of 100.

10 hops. [9] Then, by repeating the procedure of randomly picking up two points within this circle and computing the distance between them, we obtain the distribution function of H in a numerical way. The distribution $f(h)$ is shown in Fig. 10.

The other important step is to estimate p_f. We assume that $p_{f_{mo}} \gg p_{f_c}$, so $p_{f_{mo}}$ is directly used to approximate p_f. The estimation of $p_{f_{mo}}$ is done by simulation with *ns-2*. Since this parameter is determined by both movement and traffic pattern, we apply the same movement scenario as to the simulation for our protocol with the heaviest traffic load. The heaviest load of our protocol is when the network is loaded with about $F \times n$ connections and the sending rate is the basic rate imposed by the upper layer times the τ_q. For example, we simulate a scenario of 50 sources and 150 connections for a group of 50 members with $F = 3$. The results are average packet loss ratio p_l and the distribution of the number of hops H_l traveled by a packet before getting dropped (see Fig. 11 for an example). It is easy to see that $\Pr\{H_l = 1\}p_l = \sum_h p_{f_{mo}} \Pr\{H = h\}$. In fact, both sides of the equation give the probability that a packet gets lost at the first hop. Therefore, we have $p_{f_{mo}} = \Pr\{H_l = 1\}p_l$. An example of the values used for the analysis is provided in Fig. 12. It can be observed that $p_{f_{mo}}$ is an increasing function of both F and τ_q.

[9] This means that a node at the end of the diameter should take 10 hops to reach a node at the other end. The uniform distribution also implies that the path length between two nodes is approximately the same as the distance between them.

References

[1] V. Hadzilacos, S. Toueg, Fault-tolerant broadcasts and related problems, in: Distributed Systems, Addison-Wesley, Reading, MA, 1993, pp. 97–145 (Chapter 5).
[2] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, L. Zhang, A reliable multicast framework for light-weight sessions and application level framing, IEEE/ACM Transactions on Networking 5 (6) (1997) 784–893.
[3] K.P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer Systems 17 (2) (1999) 41–88.

[4] S.E. Deering, D.R. Cheriton, Multicast routing in datagram internetworks and extended LANs, ACM Transactions on Computer Systems 8 (2) (1990) 85–110.
[5] E.M. Royer, C.E. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999, pp. 207–218.
[6] S.J. Lee, M.Gerla, C.C. Chiang, On-demand multicast routing protocol, in: Proceedings of IEEE Wireless Communications and Networking Conference, 1999 (WCNC 1999), vol. 3, 1999, pp. 1298–1302.
[7] E. Pagani, G.P. Rossi, Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks, Mobile Networks and Applications 4 (3) (1999) 175–192.
[8] S.K.S. Gupta, P.K. Srimani, An adaptive protocol for reliable multicast in mobile multi-hop radio networks, in: IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 111–122.
[9] J.-P. Hubaux, T. Gross, J.-Y. Le Boudec, M. Vetterli, Toward self-organized mobile ad hoc networks: the terminodes project, IEEE Communications Magazine 39 (1) (2001) 118–124.
[10] D.B. Johnson, D.A. Maltz, Y.-C. Hu, J.G. Jetcheva, The dynamic source routing protocol for mobile ad hoc networks (DSR), February 2002, Internet-Draft, draft-ietf-manet-dsr-07.txt. Work in progress.
[11] K. Fall, K. Varadhan, (Eds.), The ns Manual, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, April 2002. Available from <http://www.isi.edu/nsnam/ns/>.
[12] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network 13 (6) (1999) 24–30.
[13] S. Čapkun, L. Buttyán, J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, IEEE Transactions on Mobile Computing 2 (1) (2003) 52–64.
[14] S. Čapkun, J.-P. Hubaux, L. Buttyán, Mobility helps security in ad hoc networks, in: Proceedings of MobiHoc'03, 2003.
[15] S. Paul, K.K. Sabnani, J.C. Lin, S. Bhattacharyya, Reliable multicast transport protocol, IEEE Journal on Selected Areas in Communications 15 (3) (1997) 784–893.
[16] M.-J. Lin, K. Marzullo, Directional gossip: gossip in a wide area network, in: Proceedings of European Dependable Computing Conference (EDCC-3), 2000.
[17] P. Eugster, R. Guerraoui, S. Handurukande, A.M. Kermarrec, P. Kouznetsov, Lightweight probabilistic broadcast, ACM Transactions on Computer Systems 21 (4) (2003) 341–374.
[18] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, Epidemic algorithms for replicated database maintenance, in: Proceedings of 6th Annual ACM Symposium on Principles of Distributed Computing (PODC'87), 1987, pp. 1–12.
[19] A.-M. Kermarrec, L. Massoulie, A. Ganesh, Probabilistic reliable dissmination in large-scale systems, IEEE Transactions on Parallel and Distributed Systems 14 (3) (2003) 248–258.
[20] R. Chandra, V. Ramasubramanian, K. Birman, Anonymous gossip: improving multicast reliability in mobile ad-hoc networks, in: Proceedings of 21st International Conference on Distributed Computing Systems (ICDCS), 2001, pp. 275–283.
[21] W. Heinzelmann, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999, pp. 174–185.
[22] S.Y. Ni, Y.C. Tseng, Y.S. Chen, J.P. Sheu, The broadcast storm problem in a mobile ad hoc network, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 1999, pp. 151–162.
[23] Z.J. Haas, J.Y. Halpern, L. Li, Gossip-based ad hoc routing, in: Proceedings of INFOCOM 2002, 2002.
[24] A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, Technical Report CS-2000-06, Duke University, 2000.
[25] L. Ji, M.S. Corson, Differential destination multicast—a MANET multicast routing protocol for small groups, in: Proceedings of INFOCOM 2001, 2001, pp. 1192–1201.
[26] K. Chen, K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in MANET, in: Proceedings of INFOCOM 2002, 2002, pp. 1192–1201.
[27] J.D. Murray, Mathematical Biology, second ed., Springer, Berlin, 1993.
[28] P. Bremaud, Markov Chains, Springer, New York, 1999.
[29] J. Luo, J.-P. Hubaux, P.Th. Eugster, PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems, in: Proceedings of MobiHoc'03, 2003, pp. 1–12.
[30] P. Eugster, R. Guerraoui, P. Kouznetsov, Delta-reliability, Technical Report DSC ID:200110, School of Computer and Communication Sciences, EPFL, 2001.
[31] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, Cluster Computing Journal 5 (2) (2002) 118–124.

Jun Luo received his B.S. and M.S. both in Electrical Engineering from Tsinghua University, Beijing, PRC, in 1997 and 2000, respectively. As a research assistant of Laboratory for Computer communication and Application (LCA), he is now working toward the Ph.D. degree in Computer Science in the Swiss Federal Institute of Technology (EPFL). His research interests include multicasting, mobile computing (especially in ad hoc networks), reliable group communication, and network security. He is a student member of ACM. For more information, check http://lcawww.epfl.ch/luo.

Patrick Th. Eugster holds a Master and a Ph.D. degree, both from the Swiss Federal Institute of Technology in Lausanne (EPFL). After having worked for some time at Chalmers University of Technology in Goteborg, Sweden, he returned to EPFL as postdoctoral researcher. His research focuses on algorithms and programming abstractions for reliable distributed systems.

Jean-Pierre Hubaux joined the faculty of the Swiss Federal Institute of Technology—Lausanne (EPFL) in 1990; he was promoted to full professor in 1996. His research activity is focused on mobile networking and computing, with a special interest in fully self-organized wireless ad hoc networks. In particular, he has performed research on cooperation aspects, security, power efficiency, and distributed algorithms for ad hoc and sensor networks.

During the last few years, he has been strongly involved in the definition and launching phases of a new National Competence Center in Research named "Mobile Information and Communication Systems" (NCCR/MICS), see http://www.terminodes.org. He served as the general chair for the Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), held in June on the EPFL campus. He is an Associate Editor of IEEE Transactions on Mobile Computing and of the Elsevier Journal on Ad Hoc Networks. Within his first year at EPFL, he defined the first curriculum in Communication Systems. From October 1999 until September 2001, he was the first chairman of the Communication Systems Department. He has held visiting positions at the IBM T.J. Watson Research Center and at the University of California at Berkeley. He has published more than 60 papers in the area of networking.

In a former life, he spent 10 years in France with Alcatel, where he was involved in R&D activities, mostly in the area of switching systems architecture and software. He is a senior member of the IEEE. For more information, check http://www.lcawww.epfl.ch/hubaux.